

Email Helpdesk for JIRA



User Manual

Authors

Marco Galluzzi, Natalio Sacerdote; edataconsulting S.L.U.

1. User Manual	3
1.1 Quick Start Guide	3
1.2 Introduction	7
1.2.1 Use cases	9
1.3 Installation	12
1.3.1 Jira Configuration	12
1.4 Plugin configuration	15
1.4.1 Filters	15
1.4.2 General Settings	20
1.4.3 Services	20
1.4.4 Templates	22
1.4.5 Logging	24
1.4.6 License	24
1.5 Working in JIRA with the plugin	24
1.5.1 Email Helpdesk issue panel	24
1.5.2 Issue panel automatic actions	25
1.5.3 Communicating with the customer	25
1.5.4 Escalating issues	25

User Manual

Quick Start Guide

In this section you will find a quick guide to get you started using the plugin, if you want to learn a little bit more before start, please, read the other sections of this document.

Step 1: Configure incoming mail

Configuring incoming mail will allow you to receive emails and create issues from those emails.

1. Log in as a user with the **'JIRA System Administrators'** global permission
2. Go to the administration section clicking on the **'Administration'** link on the top bar
3. Go to **'System'** -> **'Mail'**
4. Select the **'Incoming Mail'** Tab

The screenshot shows the JIRA Administration interface. The top navigation bar includes 'Dashboards', 'Projects', and 'Agile'. The main header is 'Administration' with a search bar. The left sidebar lists various configuration categories: General Configuration, Troubleshooting and Support, System Info, Instrumentation, Integrity Checker, Logging & Profiling, Scheduler Details, Audit Log, Atlassian Support Tools, Security, Roles, Global Permissions, Password Policy, User Sessions, Remember My Login, Whitelist, Issue Features, Time Tracking, Issue Linking, and User Interface. The main content area is titled 'Incoming Mail' and contains three sections: 'POP / IMAP Mail Servers' (with a message that no servers are configured and an 'Add POP / IMAP mail server' button), 'Mail Handlers' (with a message that no handlers are configured and an 'Add incoming mail handler' button), and 'Advanced Configuration' (with a section for 'Advanced mail loop detection' and a 'Messages Threshold' dropdown set to 0). At the bottom, there is a 'Whitelisted Domains' input field and a 'Save' button.

5. Add a POP/ IMAP Mail Server

The screenshot shows the 'Add POP / IMAP Mail Server' configuration form in the JIRA Administration interface. The form includes a warning message: 'The mail system property 'mail.mime.decodeparameters' is not set to true. This may result in e-mail attachments with non-Latin filenames being ignored. See the documentation for details.' The form fields are: 'Name*' (text input), 'Description' (text input), 'Service Provider' (dropdown menu), 'Protocol' (dropdown menu), 'Host Name*' (text input), 'POP / IMAP Port' (text input), 'Timeout' (text input), 'Username*' (text input), and 'Password*' (text input). Below the form are three buttons: 'Test Connection', 'Add', and 'Cancel'.

6. Add an Incoming Mail Handler, you should select the POP/ IMAP Mail Server you have just added and Helpdesk Mail Handler as the

Handler:

Mail Handler

Name*

Server

Delay
Delay between running time, in minutes.

Handler

Next Cancel

7. Your configuration should look something like this:

Administration Search JIRA admin

Projects Issues User management **System** Add-ons

General Configuration
Find More Admin Tools

TRUBLESHOOTING AND SUPPORT

System Info
Instrumentation
Integrity Checker
Logging & Profiling
Scheduler Details
Audit Log

ATLASSIAN SUPPORT TOOLS
Atlassian Support Tools

SECURITY

Roles
Global Permissions
Password Policy
User Sessions
Remember My Login
Whitelist

ISSUE FEATURES
Time Tracking

Incoming Mail

POP / IMAP Mail Servers

The table below shows the POP / IMAP mail servers currently configured for JIRA.

⚠ The mail system property 'mail.mime.decodeparameters' is not set to true. This may result in e-mail attachments with non-Latin filenames being ignored. See the documentation for details.

Name	Details	Operations
Test Incoming Mail Server	Host: pop.gmail.com Username: test.fdre	Edit · Delete

Add POP / IMAP mail server

Mail Handlers

The table below shows the mail handlers currently configured for JIRA.

Handler Name / Type	Server	Project	Issue Type	Properties	Operations
Test Mail Handler Helpdesk Mail Handler	Test Incoming Mail Server pop.gmail.com			Unknown:	Edit · Delete

Add incoming mail handler

Step 2: Configure outgoing mail

Configuring outgoing mail will allow the plugin to send automatic notification emails when an issue event is triggered.

(If you already have configured outgoing mail you can skip this step)

1. Log in as a user with the '**JIRA System Administrators**' global permission
2. Go to the administration section clicking on the '**Administration**' link on the top bar
3. Go to '**System**' -> '**Mail**'
4. Select the '**Outgoing Mail**' Tab
5. Add a SMTP Mail Server and make sure Outgoing mail is enabled.

The screenshot shows the JIRA Administration interface. The top navigation bar includes 'Dashboards', 'Projects', and 'Agile'. The main header is 'Administration' with a search bar. The left sidebar lists various administration categories: General Configuration, Troubleshooting and Support, System Info, Instrumentation, Integrity Checker, Logging & Profiling, Scheduler Details, Audit Log, Atlassian Support Tools, Security, Roles, Global Permissions, Password Policy, User Sessions, Remember My Login, WhiteList, Issue Features, Time Tracking, Issue Linking, and User Interface. The main content area is titled 'Outgoing Mail' and is marked as 'ENABLED'. It includes a button 'Disable Outgoing Mail' and a description: 'When enabled and configured, JIRA will be able to send emails to users.' Below this is the 'SMTP Mail Server' section, which states: 'The table below shows the SMTP mail server currently configured for JIRA.' A table follows with columns 'Name', 'Details', and 'Operations'. The table contains one entry: 'Test Outgoing Mail Server'. The details for this server are: From: test.fdre@gmail.com, Prefix: [JIRA], Host: smtp.gmail.com, SMTP Port: 465, and Username: test.fdre. The operations column provides links for 'Edit', 'Delete', and 'Send a Test Email'.

Step 3: Add a new Service

The Service is the place where the information needed to handle incoming emails is stored.

1. Log in as a user with the '**JIRA System Administrators**' global permission
2. Go to the administration section clicking on the '**Administration**' link on the top bar
3. Go to '**Plugins**' -> '**Support System Admin**'
4. Select the '**Services**' Tab

The screenshot shows the JIRA Administration interface for 'Email Helpdesk Configuration'. The top navigation bar is the same as the previous screenshot. The main header is 'Administration' with a search bar. The left sidebar lists various administration categories, including 'Email Helpdesk Configuration'. The main content area is titled 'Email Helpdesk Configuration'. It includes a 'Services' tab and a 'Service List' table. The 'Service List' table has columns: 'Name', 'Mail Handler', 'Reply-To Email', 'Sender Name', 'Attachments', and 'Actions'. The table is currently empty, and a message below it states: 'You do not have any service configured yet, click on Add Service to begin...'. There is an 'Add Service' button above the table.

5. Click on '**Add Service**' button
6. The New Service form will appear and you will see the following fields:
 - a. **Service ID**: An identifier for this service, the ticket number format will be ServiceID-1, ServiceID-2 and so on
 - b. **Mail Handler**: the mail handler where we will get the mails from. A mail handler is always associated with an incoming mail server, and this incoming mail server is normally associated with a mail address.
 - c. **Reply-To**: should be the same mail address associated with the mail handler
 - d. **Sender Name**: the name that will appear as the sender of the notification emails sent to the user that originally created the issue.
 - e. **Add attachments**: If 'Yes' is selected attachments received from the user will be attached to the issue
 - f. **Max Size**: If attachments are enabled, you can specify here the maximum attachment size

Administration Search JIRA admin

Projects Issues User management System **Add-ons**

ATLASSIAN MARKETPLACE
Find new add-ons
Manage add-ons
Purchased add-ons
OSGI

APPLICATION LINKS
Application Links

SOURCE CONTROL
DVCS Accounts
FishEye Configuration
Perforce Job Integration

BUILDS
Bamboo configuration

ISSUE COLLECTORS
Issue Collectors

EMAIL HELPDESK
Configuration

MONITORING
Database Monitoring

ADMIN HELPER
Permission Helper
Notification Helper

Email Helpdesk Configuration

Services

Filters
Templates
General
Log
License

New Service

Service ID* Test Service

Mail Handler* Test Mail Handler

Reply-To Email* test.fdre@gmail.com

Sender Name* Support Test

Add attachments ☒ Yes ☐ No

Max Size (MB)* 10

First Level Project

[Add](#)

Project Key	Assignee	Reporter	Show Email	Default action on comment	Actions
You must add at least one First Level Project					

Second Level Projects

[Add](#)

Project Key	Assignee	Show Email	Default action on comment	Actions
No Data				

Third Level Projects

[Add](#)

Project Key	Assignee	Show Email	Default action on comment	Actions
No Data				

[Save](#) [Cancel](#)

7. You must add a First Level Project, a First Level Project is required in order to get the issues created with the specified options, you will be asked for:
- Project Key:** The key of the project where we want the issues to be created
 - Assignee:** The default user the created issues will be assigned to
 - Reporter:** The user we want to be the reporter of the new issues, this user will be also the author of the external comments received.
 - Show Email:** If 'Yes' is selected, the user email will be displayed in the view issue panel
 - Default action comments:** If 'Send' is selected, when we comment an issue this comment will be send to the user by default, we should specify that is an internal comment in order to not send it

Add Project to First Level

Project Key* TEST

Assignee* supportagent

Reporter internetusers

Show Email ☒ Yes ☐ No

Default action on comment ☒ Send ☐ Do not send

[Add](#) [Cancel](#)

8. Now you can save the Service and you will be redirected to the Service list page:

Administration Search JIRA admin [Back to project: TEST](#)

Projects Issues User management System **Add-ons**

ATLASSIAN MARKETPLACE
Find new add-ons
Manage add-ons
Purchased add-ons
OSGI

APPLICATION LINKS
Application Links

SOURCE CONTROL
DVCS Accounts
FishEye Configuration
Perforce Job Integration

BUILDS
Bamboo configuration

ISSUE COLLECTORS
Issue Collectors

EMAIL HELPDESK
Configuration

MONITORING
Database Monitoring

ADMIN HELPER
Permission Helper
Notification Helper

Email Helpdesk Configuration

Services

- Filters
- Templates
- General
- Log
- License

Service List

[Add Service](#)

Name	Mail Handler	Reply-To Email	Sender Name	Attachments	Actions
Test Service	Test Mail Handler	test.fdre@gmail.com	Support Test	true	Edit Delete

? In order to be able to add a Service, you need to add at least one Helpdesk Mail Handler first.
Click on [Create Helpdesk Mail Handler](#) to create one.

Step 4: Test it

In this step we will test the plugin and learn the basics of how to use it:

1. Send an email to the address of the POP/ IMAP Mail Server you have configured in the previous steps.
2. A new issue will be created in the project specified in the "First Level Project" and you should receive an email confirmation (It may take a while)
3. Open the new issue
4. You will see on the right side a Panel named '**Support System Jira Plugin**', with the fields being detailed below:
 - a. **Ticket #:** The ticket number associated with this issue, you can also find it labelled to the issue
 - b. **Default Action:** The default action when commenting this issue. (Send / Don't Send the comment as an email to the user who created the ticket, the one who sent the email in the first step)
 - c. **Expression to Send/not Send:** The expression we should start our comment with in order to be sent/not be sent as an email to the user who created the ticket (the one who sent the email in the first step). Examples: "I: this is an internal comment", "Q: this is a public comment"
 - d. **User email:** user mail address that originated this ticket or 'XXXXXX' depending on the option chosen when the Service was configured, specifically the 'Show email' option of the First Level Project.
 - e. **Ticket Level:** The current support level of this ticket, when the plugin creates a ticket it's automatically assigned to the First Level. The possible values are: First Level (1), Second Level (2) or Third Level (3).
 - f. **Link Escalate/De-Escalate issue:** This links should be used to move the issue through the support levels.
5. Check the **Default Action** and the **Expression to Send/not Send** in the Panel and then write a public comment in the issue. You should receive an email notification (sent to the mail address of the user who originated this issue) with the comment. (It may take a while)
6. Reply to the email you have just received, make sure that the Ticket # is on the subject (This is needed in order to identify the ticket). You should see your reply email added as a comment to the issue. You should receive an email confirmation. (It may take a while)
7. Done! You have finished testing your configuration!

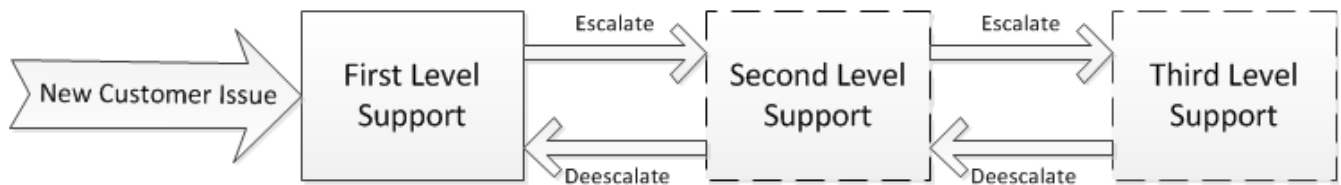
Introduction

Overview

The aim of Email Helpdesk for JIRA is to improve and configure JIRA as an email support system. Email Helpdesk enables you to use JIRA as an email based helpdesk system by creating and commenting on issues based on received emails and vice versa (sending emails on comments).

In order to serve a business or customer base efficiently, technical support is often organized into levels. First level support is normally responsible of handling basic customer issues while higher support levels offer a more in-depth technical support. Email Helpdesk is able to handle up to three support levels (first support level is mandatory). When a customer issue is received it will be assigned to the first level support, if the first level support is not able to handle the issue it will escalate the issue to the next support level.

A support level is defined as a set of projects, except for the First Level Support that always consists of just one project.



Main Features

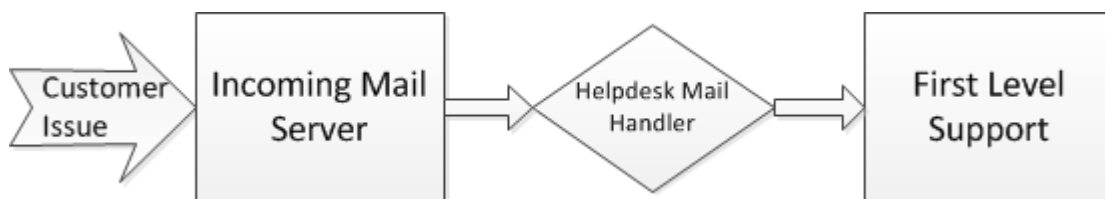
These are the Email Helpdesk main features:

1. Allow any user to be able to interact with JIRA, including users without a JIRA account.
2. Filter received emails
3. Automatically notify external users when issue is resolved, created or commented
4. Issue escalating/de-escalating
5. Communication between support and user through issue comments
6. Customizable notification emails

Incoming Mail Handling

Incoming Mail Server

In order to be able to receive emails in Jira, an incoming mail server needs to be configured and a Mail Handler assigned to it is also needed. The Incoming mail server has the information to retrieve the emails and the Mail Handler determines what to do with them. Email Helpdesk provides a Mail Handler (Helpdesk Mail Handler). Helpdesk Mail Handler will create or comment issues based on the received emails.



Service

When an email from a customer is received and the Helpdesk Mail Handler is about to create or comment an issue from this email, it needs some information to perform this task, for example, in which project the issue will be created or who will be the assignee. This information is stored in the Service.

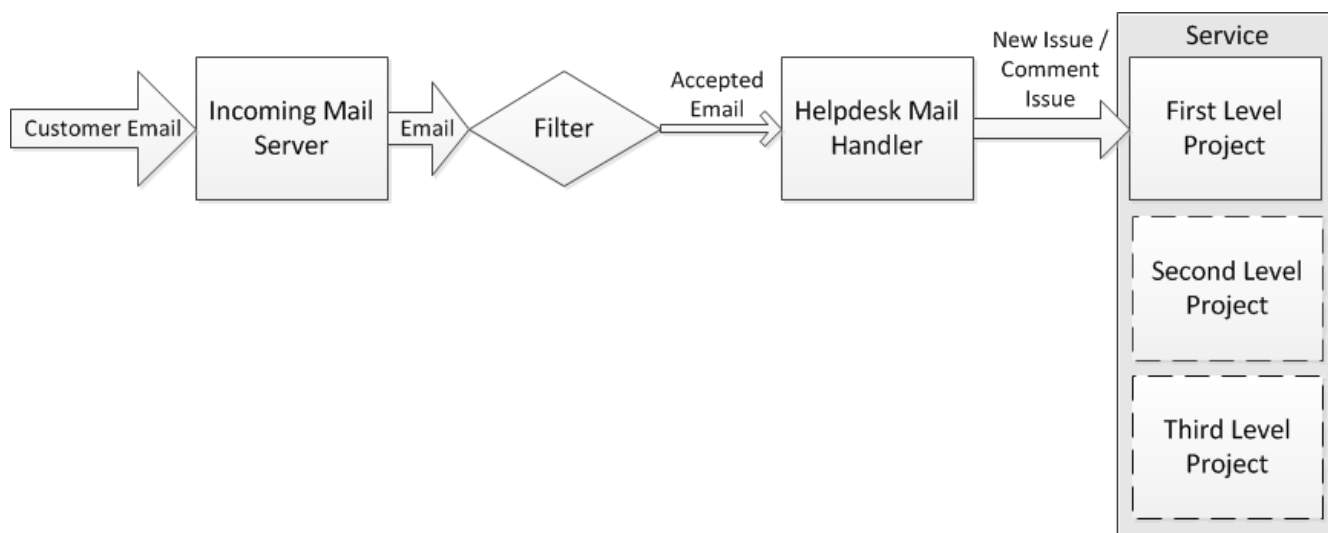
Filters

Filters are used to reject emails based on rules that we can set up (it is also possible use them to accept emails); we will talk about them later on this document. Refer to the [Filters](#) section.

Process summary

When an email is received, the plugin first checks if the email passes the defined filter (note that defining a filter is optional), the plugin proceeds to check if it should create a new issue or comment an existing one.

When the plugin creates an issue, the issue will have some extra information associated to it, it will have a ticket number and the email address of the user who sent the email.



Outgoing Mail Server

An Outgoing Mail Server is needed to send notifications to external users, for instance, when an issue created by the plugin is resolved or commented, an email will be sent to the user that originally created the issue.

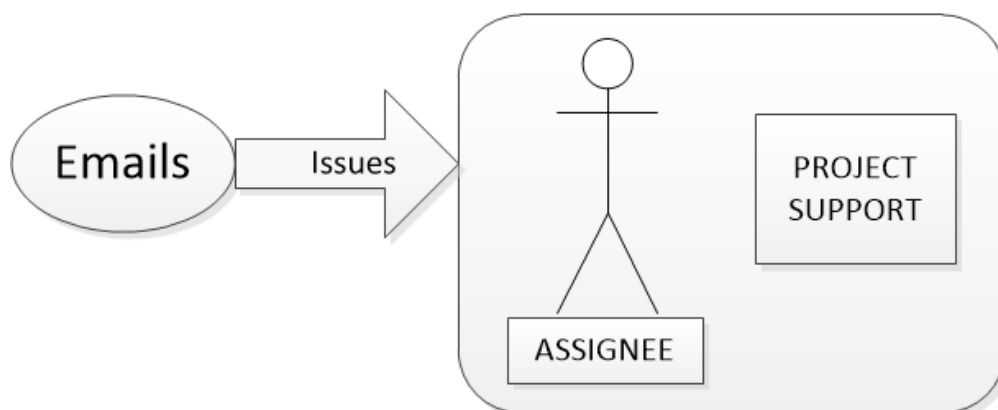


Use cases

Different scenarios and configurations will be described In this section as examples that might be useful.

Simple Use Case

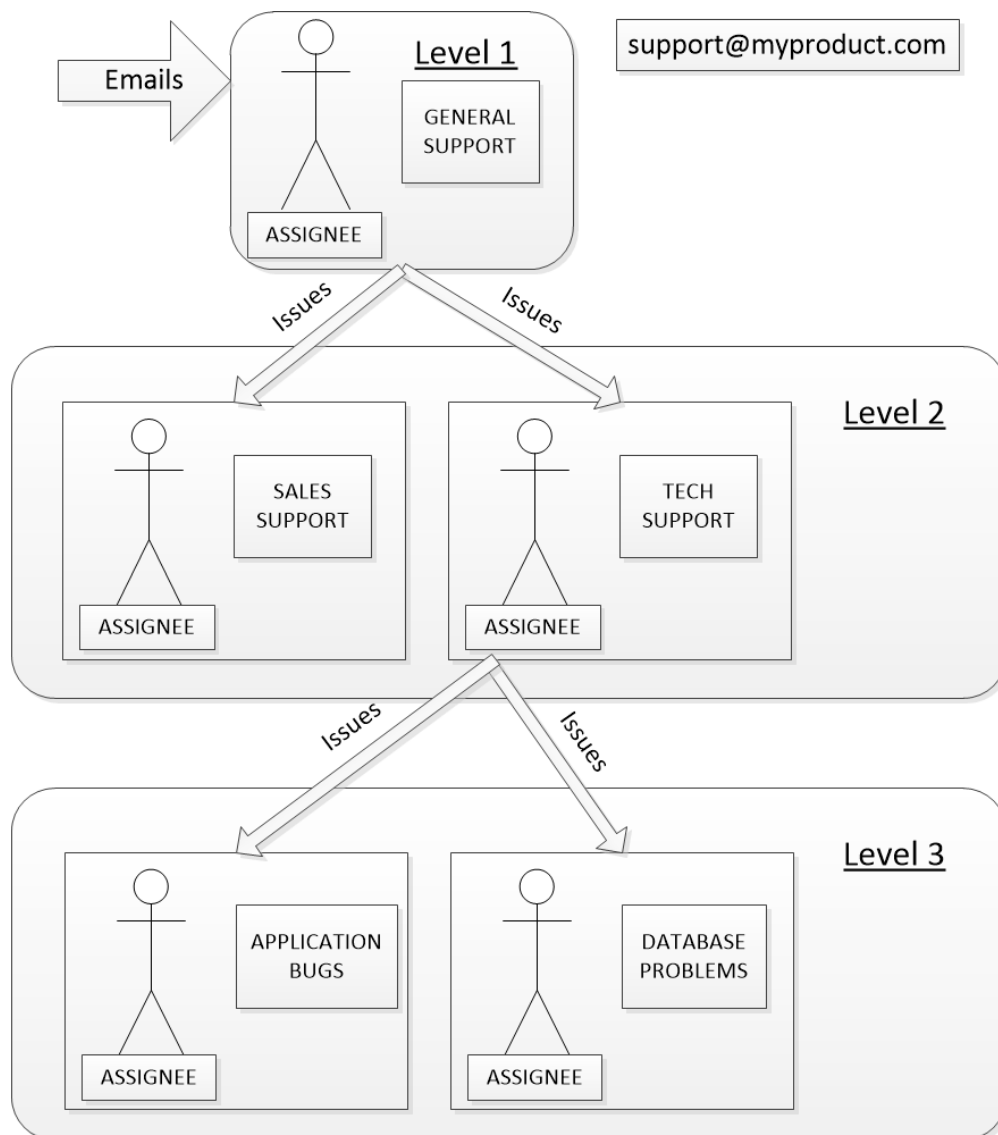
support@myproduct.com



In this example we have configured an Incoming Mail Server to listen on address support@myproduct.com (Check [Jira Configuration](#) to learn how to do this). We also have configured one first level project in our service (Visit [Services](#) section to learn more). With this configuration all emails received on support@myproduct.com will be created as issues in the project "SUPPORT" and assigned to the user "ASSIGNEE" who will be the responsible of handling user's tickets. ([Example screenshot configuration](#))

This scenario is useful when we like to keep our support system as simple as possible.

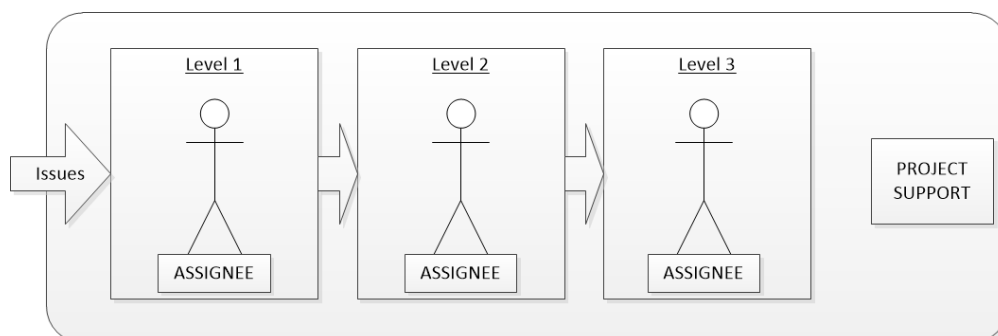
Complex Use Case



In this example we have configured an Incoming Mail Server to listen on address support@myproduct.com. We also have configured one first level project, two second level projects and two third level projects in our service. With this configuration all emails received on support@myproduct.com will be created as issues in the project "GENERAL SUPPORT". The assignee of this project will be the responsible of handling user's tickets, if he receives a ticket that he can't handle, he will have to decide if the ticket is for the sales support team or for the technical support team and escalate appropriately. ([Example screenshot configuration](#))

This scenario is useful when we have a complex product and we need to provide different levels of support.

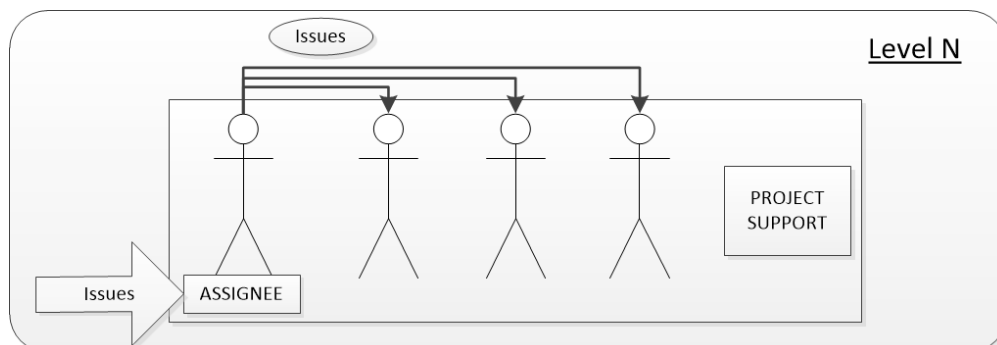
1 Project 3 Levels Use Case



In this example we have configured our service with three levels of support, on every level we have used the same project but we configured a different assignee per level ([Example screenshot configuration](#)). All the tickets received will be assigned to the first level assignee, and then he can decide to escalate this issues to the next level, the issue will remain in the same project, but the assignee will change.

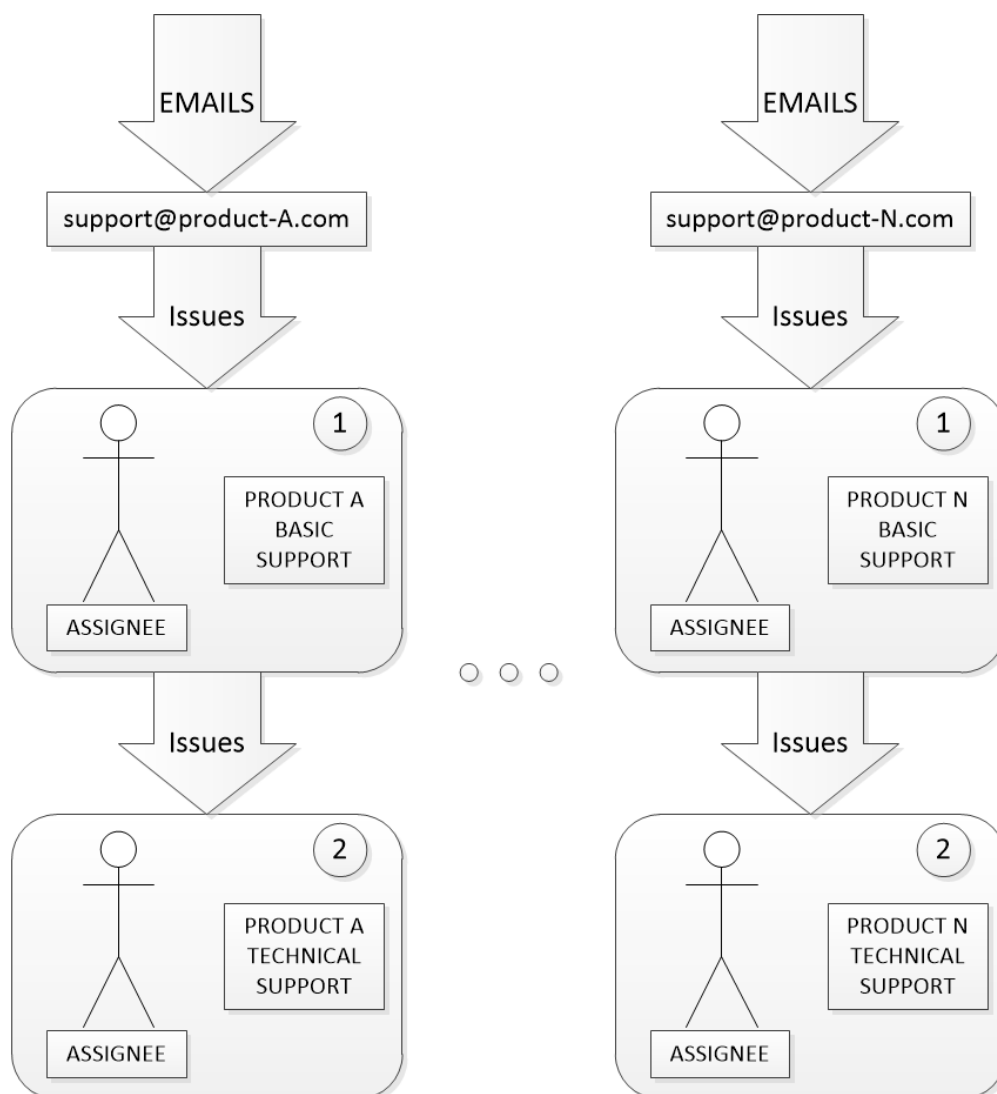
This scenario is useful if we want to have different levels of support but we only want to use one project for this support system.

N people 1 Project Use Case



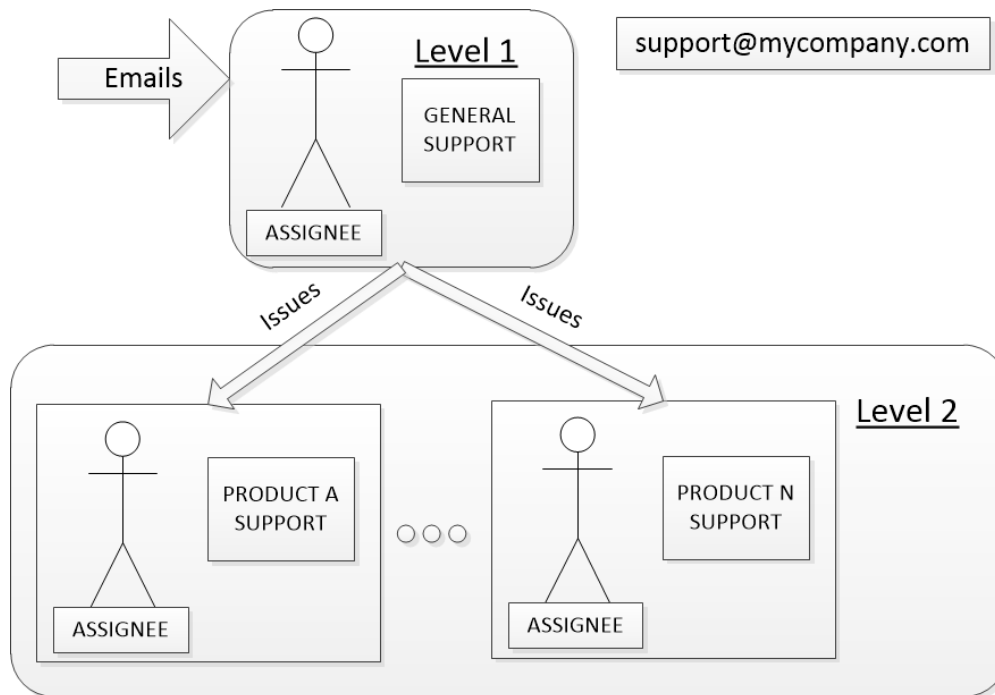
Here we have an example of how we could work if we want many people to work on the same level of support. We should make one of them the default assignee of the received issues, then he will be the responsible of the work distribution to the other employees (assign the issue). ([Example screenshot configuration](#))

N products N Services Use Case



When we have many products and we want to provide support for them we can choose a configuration like this one. We configure as many incoming mail servers as products we want to support and then we configure a service for each of this incoming mail servers. ([Example screenshot configuration](#))

N products 1 Service Use Case



If we want to support many products but we just want to use one email address, we can configure the plugin to work like this example. One incoming mail server must be configured and we must create one service for it, in the first level project we will have a general support project and in the second level we will have specific support for each of our products. When we receive tickets in "support@mycompany.com" tickets will be created in the general support project, and there the assignee should decide if he escalates the issue to an specific product support project in level 2. ([Example screenshot configuration](#))

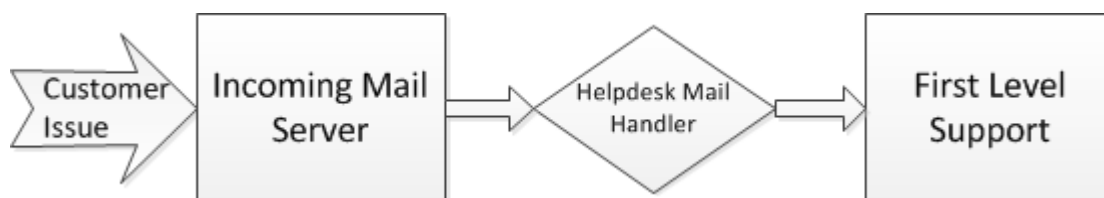
Installation

Jira Configuration

Some changes in Jira Configuration are needed in order to get EHJ working properly.

Incoming Mail Server

As already mentioned, in order to be able to receive emails in Jira, an incoming mail server needs to be configured and a Mail Handler assigned to it is also needed. The Incoming mail server has the information to retrieve the emails and the Mail Handler determines what to do with them. EHJ provides a Mail Handler (Helpdesk Mail Handler).



Configure Incoming Mail Server

The incoming mail server configuration tells Jira how to connect and retrieve emails from our email account. To configure an Incoming Mail Server, follow these steps:

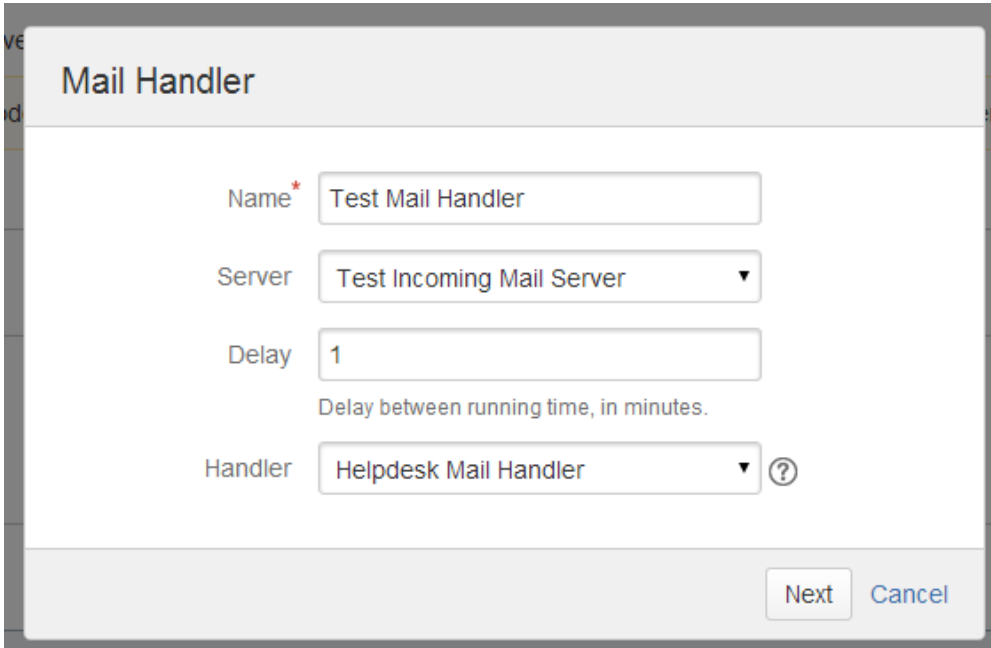
1. Log in as a user with the '**JIRA System Administrators**' global permission
2. Go to the administration section clicking on the '**Administration**' link on the top bar
3. Go to '**System**' -> '**Mail**'
4. Select the '**Incoming Mail**' Tab

5. Add a POP/ IMAP Mail Server. [Learn More](#)

Configure a Mail Handler

The Helpdesk Mail Handler provided with the plugin is the responsible of handling the emails retrieved by the Incoming Mail Server. In order to be able to add a [Service](#) we need to set up a Helpdesk Mail Handler. To configure a Mail Handler, follow these steps:

1. Log in as a user with the 'JIRA System Administrators' global permission
2. Go to the administration section clicking on the 'Administration' link on the top bar
3. Go to 'System' -> 'Mail'
4. Select the 'Incoming Mail' Tab
5. Add an Incoming Mail Handler, you should select the POP/ IMAP Mail Server you have just added and Helpdesk Mail Handler as the Handler:



The image shows a 'Mail Handler' configuration window. It has a title bar 'Mail Handler'. Inside, there are four fields: 'Name' with a red asterisk and the value 'Test Mail Handler'; 'Server' with a dropdown menu showing 'Test Incoming Mail Server'; 'Delay' with a text input '1' and a note 'Delay between running time, in minutes.'; and 'Handler' with a dropdown menu showing 'Helpdesk Mail Handler' and a help icon. At the bottom right are 'Next' and 'Cancel' buttons.

** Note that if you delete a Helpdesk Mail Handler associated with a Service, it will stop working until you select a new Helpdesk Mail Handler for it.*

Outgoing Mail Server

An Outgoing Mail Server is needed to send notifications to external users, for instance, when an issue is resolved or commented, an email will be sent to the user that originally created the issue.



Configure outgoing mail

1. Log in as a user with the '**JIRA System Administrators**' global permission
2. Go to the administration section clicking on the '**Administration**' link on the top bar
3. Go to '**System**' -> '**Mail**'
4. Select the '**Outgoing Mail**' Tab
5. Add a SMTP Mail Server and make sure Outgoing mail is enabled.

The screenshot shows the JIRA Administration interface. The top navigation bar includes 'JIRA', 'Dashboards', 'Projects', and 'Agile'. The main header is 'Administration' with a search bar. The left sidebar lists various administration categories: General Configuration, Troubleshooting and Support, Atlassian Support Tools, Security, Issue Features, and User Interface. The main content area is titled 'Outgoing Mail' and is currently 'ENABLED'. It includes a 'Disable Outgoing Mail' button and a note that JIRA will be able to send emails to users when enabled. Below this, the 'SMTP Mail Server' section shows a table with the current configuration for JIRA.

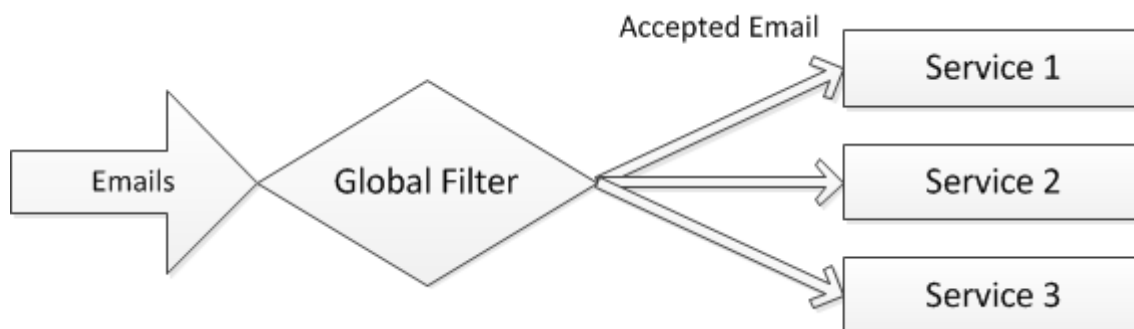
Name	Details	Operations
Test Outgoing Mail Server	From: test.tdree@gmail.com Prefix: [JIRA] Host: smtp.gmail.com SMTP Port: 465 Username: test.tdree	Edit Delete Send a Test Email

Plugin configuration

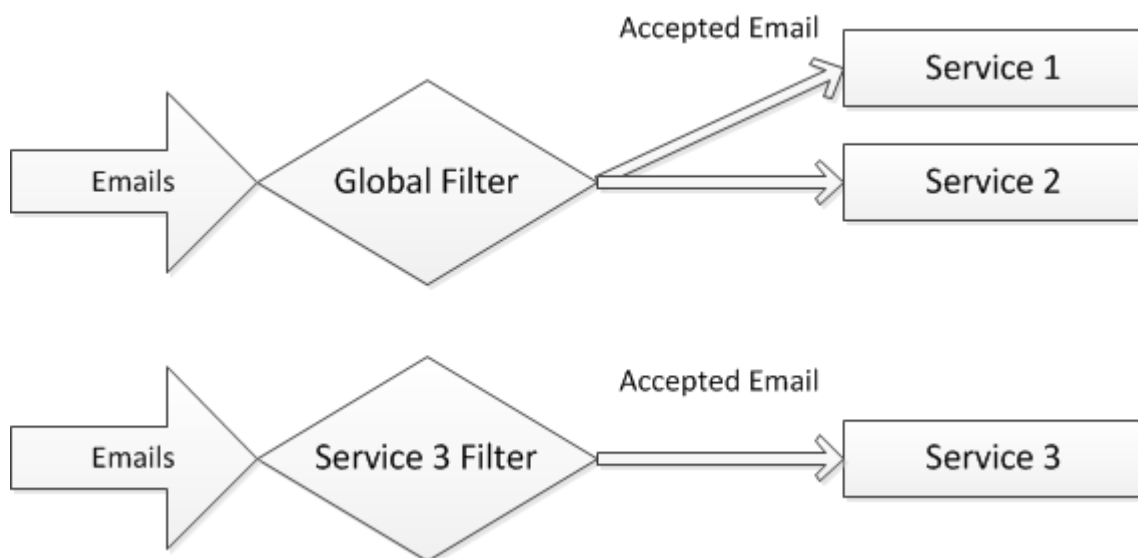
Filters

Filters are used in order to reject emails based on user defined criteria; the plugin gives you the option to set up a service specific filter or a global filter.

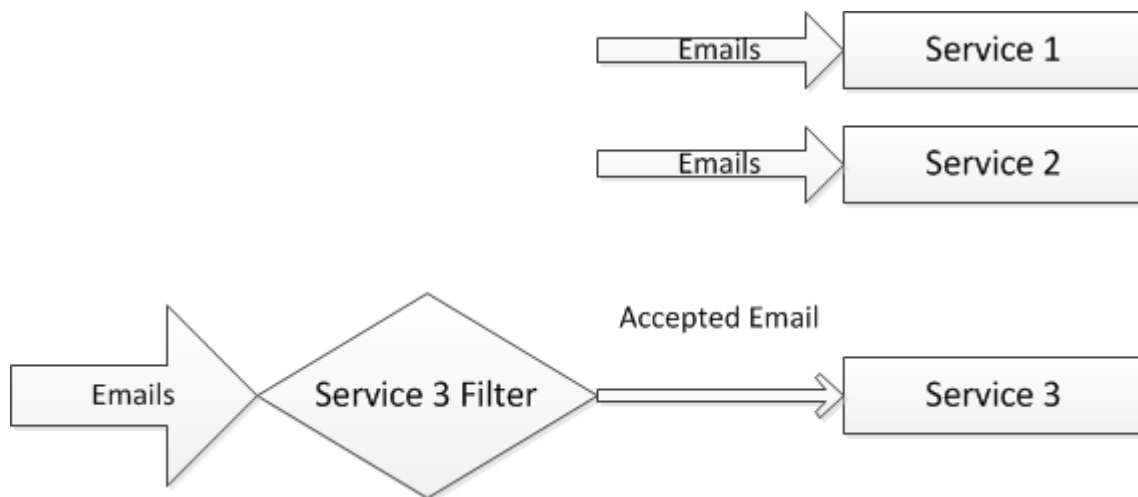
When only a global filter is set, every email go through this filter:



If a global filter is combined with a service specific filter, emails for this service are handled by the specific filter and the other emails go through the global filter:



If only service specific filters are configured, services without a specific filter defined will get the emails directly:



Creating a Filter

Email Helpdesk for Jira Configuration

New Filter

Scope:

Deny

Reject emails that match of the following conditions

Conditions Add

Rule	Apply To	Expression	Actions
No Data			
Blacklist		<div></div>	

Allow

Allow emails that match of the following conditions

Conditions Add

Rule	Apply To	Expression	Actions
No Data			
Whitelist		<div></div>	

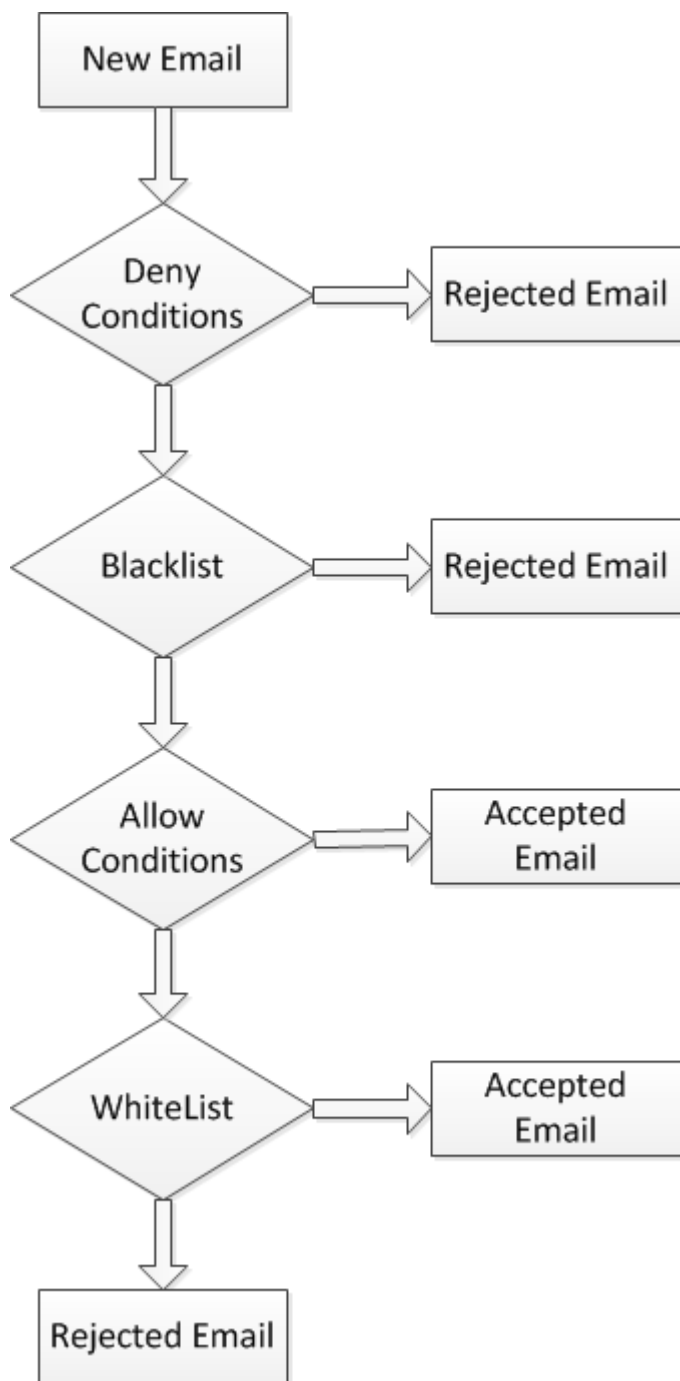
The following options are available when creating a filter:

- **Scope:** There are two options, a global filter or a specific filter. A global filter will be applied to any Service without a specific filter. A specific filter will be only applied to the associated Service. Note that you can only have one global filter and one specific filter per service.
- **All / Any:** When 'All' is selected the received emails will have to meet all the Deny conditions in order to be rejected. If 'Any' is selected, the email will have to meet just one of the Deny conditions in order to be rejected.
- **Deny Conditions:** List of conditions, a condition consists of:
 - Name: An identifier for the condition
 - Apply to: Three options are available:
 - 'Sender / From': this condition will be applied to the sender email address
 - 'Subject': this condition will be applied to the subject of the email
 - 'Body': this condition will be applied to the body of the email
 - Expression: A Java regular expression. [Learn more](#)
- **Blacklist:** mail address list that have to be blocked. The semicolon is used as separator.
- **All / Any:** When 'All' is selected the email will have to meet all the Allow conditions in order to be accepted. If 'Any' is selected, the email will have to meet just one of the Allow conditions in order to be accepted.
- **Allow Conditions:** List of conditions, a condition consists of:
 - Name: An identifier for the condition
 - Apply to: Three options are available:
 - 'Sender / From': this condition will be applied to the sender email address
 - 'Subject': this condition will be applied to the subject of the email
 - 'Body': this condition will be applied to the body of the email
 - Expression: A Java regular expression. [Learn more](#)
- **Whitelist:** mail address list that have to be accepted. The semicolon is used as separator.

Logic

There are four possible ways to set up a Filter, with deny and allow rules, with only deny rules, with only allow rules, or an empty filter. The logic used when a filter is being applied will be explained below.

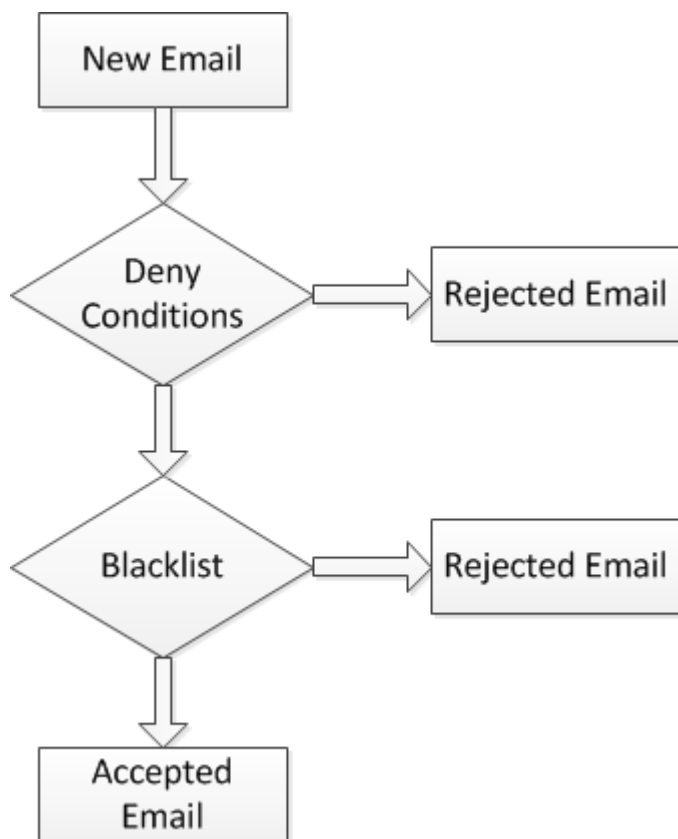
Deny and Allow



If the **Deny** and **Allow** sections of the filter are configured:

1. The plugin will check the Deny rules and the blacklist, reject the email if meets the deny conditions or the sender is blacklisted.
2. The plugin will check the Allow rules and whitelist, accept the email if meets the allow conditions or the sender is whitelisted.
3. The plugin will reject the email because it was not allowed by the allow conditions / whitelist

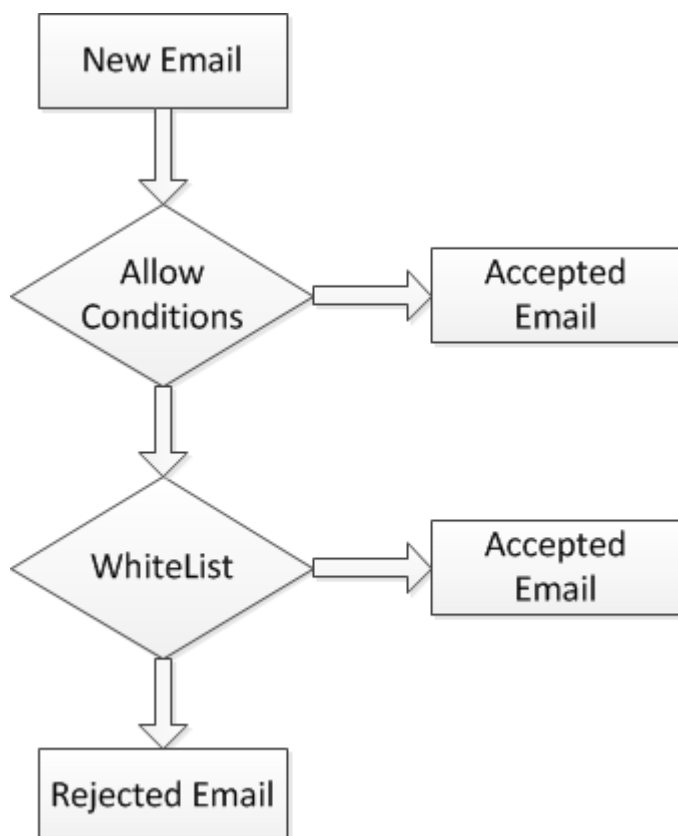
Only Deny



If **only the Deny** section of the filter is configured

1. The plugin will check the Deny rules and blacklist, reject the email if meets the deny conditions or the sender is blacklisted.
2. The plugin will accept the email because it wasn't rejected by the deny conditions / blacklist

Only Allow



If **only the Allow** section of the filter is configured:

1. The plugin will check the Allow rules and whitelist, accept the email if meets the allow conditions or the sender is whitelisted.
2. The plugin will reject the email because it wasn't allowed by the allow conditions / whitelist

Empty Filter

If an **empty filter** is created, the emails will always pass the filter. This can be useful when we want a service to get all incoming messages without the global filter interfering,

General Settings

There are some extra options that can be configured in the General Settings section:

- Reject automatically generated emails: If this option is enabled the plugin will try to determine if an email was generated automatically and reject it.
- HTML emails / Text emails: outgoing mail format, if HTML selected you are allowed to insert html tags in the Templates section.
- Internal / External Comment Mark: Expression used to specify whether a comment is Internal or External

Email Helpdesk Configuration

The screenshot shows the 'General Settings' dialog box for the Email Helpdesk Configuration. On the left is a sidebar with navigation links: Services, Filters, Templates, General (selected), Log, and License. The main panel contains the following settings:

- ☒ Reject automatically generated emails
- Email format: ☐ HTML emails ☒ Text emails
- Private Comment Mark*: PRIV:
- Public Comment Mark*: PUB:

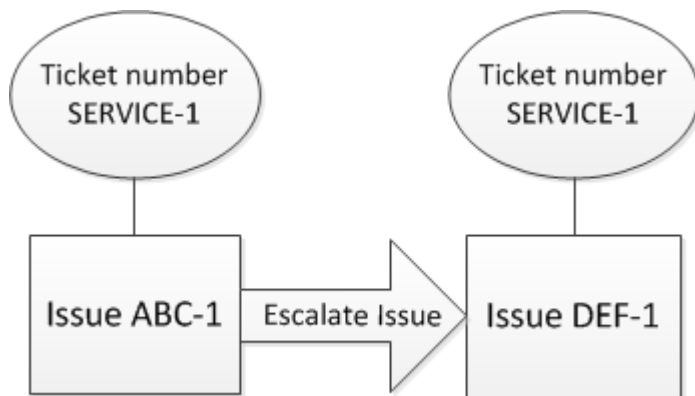
At the bottom are three buttons: Save, Default values, and Cancel.

Services

When an email from a customer is received and the Helpdesk Mail Handler is about to create an issue from this email, it needs some information to perform this task, for example, in which project the issue will be created, who will be the assignee or if the attachments should be added to the issues. This information is stored in the Service.

Ticket Number

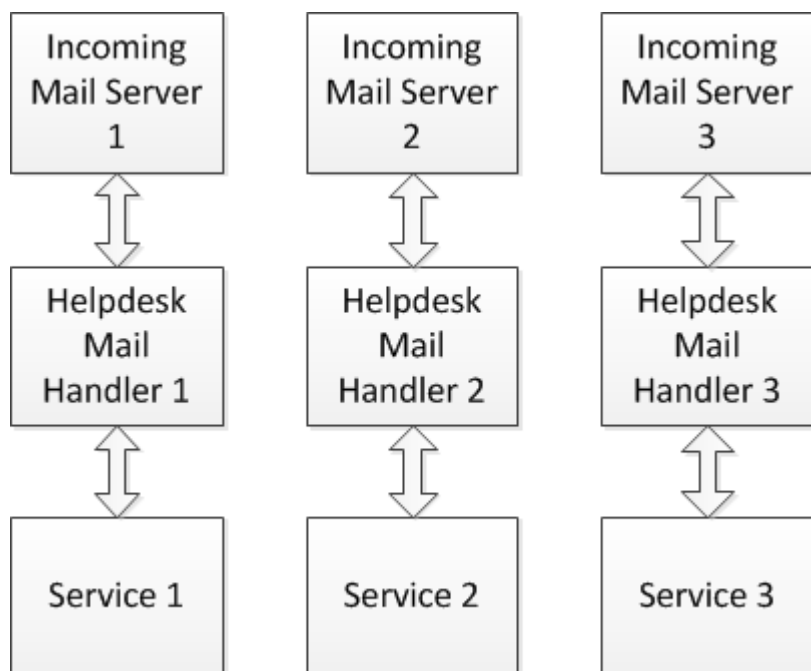
When an issue is created by the plugin a ticket number is assigned to it, this ticket number helps identifying this issue even if we move it across different projects.



Service Relation

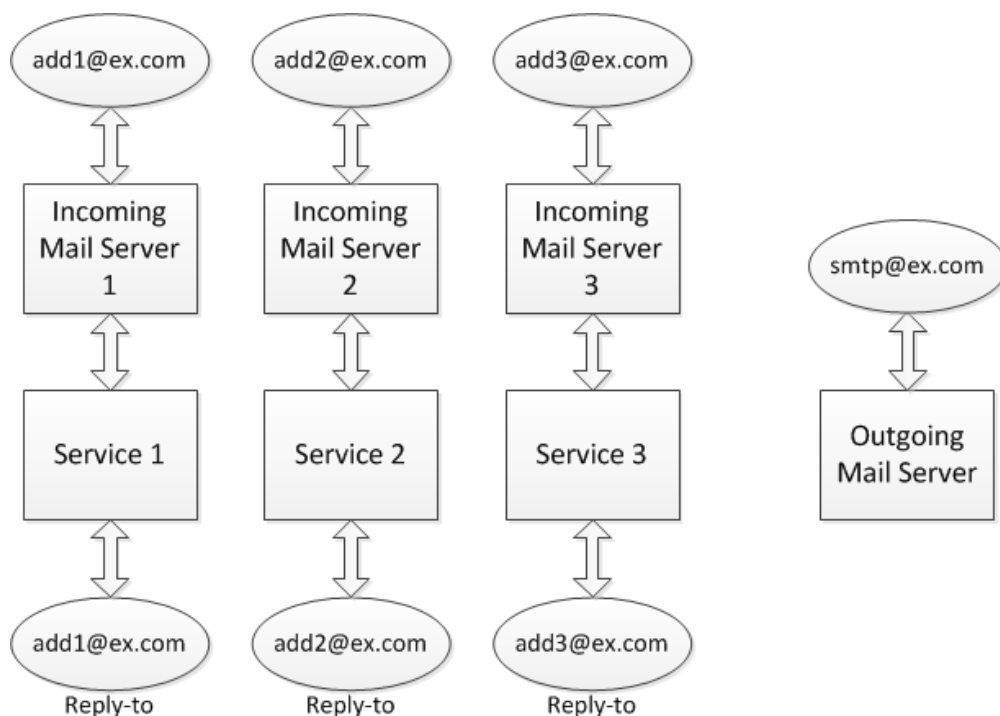
A Service needs to be always associated with a Helpdesk Mail Handler in order to properly work. We can have more than one Service; all of them should be associated with a Mail Handler.

* Note that if we didn't create any Helpdesk Mail Handler or there are no available (all assigned to a service already) Helpdesk Mail Handlers then we can't add a new service. If you want to learn how to add a Helpdesk Mail Handler, please refer to the [Jira Configuration Section](#).



Reply-To

Jira only allows one Outgoing Mail Server to be configured, that's why the plugin needs a Reply-To field:



When the plugin sends an email it's sent from the only Outgoing Mail Server available and from the associated address (smtp@ex.com in the example), in order to get the reply emails in the correct inbox (add1@ex.com, add2@ex.com or add3@ex.com) the plugin allows you to configure a reply-to address. If you configure a reply-to address different from the one associated with the selected mail handler your customers replies will not be received appropriately.

Reporter

Normally it's good to have a separate user for being the reporter of the issues created by the plugin, this user will also be used to comment the issues. As a recommendation, in order to be able to identify the actions performed by the plugin quickly, you could name the user: customers, externalUser or emailUser.

Creating or Editing a Service

These are the fields stored in a Service:

Administration Search JIRA admin

Projects Issues User management System Add-ons

ATLASIAN MARKETPLACE
Find new add-ons
Manage add-ons
Purchased add-ons
OSGi

APPLICATION LINKS
Application Links

SOURCE CONTROL
DVCS Accounts
FishEye Configuration
Perforce Job Integration

BUILDS
Bamboo configuration

ISSUE COLLECTORS
Issue Collectors

EMAIL HELPDESK
Configuration

MONITORING
Database Monitoring

ADMIN HELPER
Permission Helper
Notification Helper

Email Helpdesk Configuration

New Service

Services

Filters

Templates

General

Log

License

Service ID*

Mail Handler*

Reply-To Email*

Sender Name*

Add attachments ☒ Yes ☐ No

Max Size (MB)*

First Level Project Add

Project Key	Assignee	Reporter	Show Email	Default action on comment	Actions
You must add at least one First Level Project					

Second Level Projects Add

Project Key	Assignee	Show Email	Default action on comment	Actions
No Data				

Third Level Projects Add

Project Key	Assignee	Show Email	Default action on comment	Actions
No Data				

- **Service ID:** An identifier for this service. Note that ticket number format will be: *ServiceID-1*, *ServiceID-2*, and so on.
- **Mail Handler:** The mail handler associated with this Service. Only Helpdesk Mail Handlers that are not already selected in other service will be displayed in the list. If you want to learn how to add a Helpdesk Mail Handler, please refer to the [Jira Configuration Section](#). If we delete the Service, the selected mail handler will be again available to be selected by the other services.
- **Reply-To:** It should be the same mail address associated with the mail handler. If you configure a reply-to address different from the one associated with the selected mail handler your customers replies will not be received appropriately.
- **Sender Name:** The name that will appear as the sender of the emails sent to the user that originally created the issue.
- **Add attachments:** If 'Yes' is selected attachments received from the user will be attached to the issue.
- **Max Size:** If attachments are enabled, you can specify here the maximum attachment size.
- **First Level Project:** The project where the issues will be created automatically when an email is received, here it's also stored some information about the plugin behaviour:
 - **Project Key:** The key of the project where we want the issues to be created.
 - **Assignee:** The default user the created issues will be assigned to.
 - **Reporter:** The user we want to be the reporter of the new issues, this user will be also the author of the external comments received.
 - **Show Email:** If 'Yes' is selected, the user email will be displayed in the view issue panel. If you want to learn more about this panel, please visit the [Using the Plugin Section](#).
 - **Default action comments:** If 'Send' is selected, when we comment an issue this comment will be sent to the user by default, we should specify that is an internal comment in order to not send it. If you want to learn more about how to specify public / internal comments, please visit the [Using the Plugin Section](#).
- You can also add Second and/or Third Level Projects; this will allow you to move the tickets between the levels using escalate/de-escalate buttons.
- Note also that only one project can be added in the First Level Project. However we can add as many projects as we want to the 2nd and 3rd Levels, with the only condition- no duplicated projects in the same Level. If you have more than one project in a Level, when escalating/de-escalating you will be asked which project you want the issue to be moved to.

In any moment we can click on Cancel so no changes will be performed.

When we click on save button then new service appears in the Service List page.

** Note that if you delete a Helpdesk Mail Handler associated with a Service, it will stop working until you select a new Helpdesk Mail Handler for it.*

Templates

Templates allow you to customize the way the received email content is used (inbound templates) and the way the email notifications are created (outbound templates).

Inbound templates

Email Helpdesk Configuration

Inbound Templates	
Issue Creation Summary * <input type="text" value="\$subject\$"/> Description * <div><div>\$body\$</div></div>	Issue Commented Comment * <div><div>\$body\$</div></div>
<p>Inbound Templates</p> <ul style="list-style-type: none"> \$subject\$ -- The subject of the incoming email \$body\$ -- The body of the incoming email 	

When an email is received there are two possible actions to perform, issue creation or comment the issue:

- **Issue Creation:** it's used when an email is received and the plugin has to create a new ticket. The available fields to customize are:
 - **Summary:** the summary of the issue that is being created
 - **Description:** the description of the issue that is being created
- **Issue Commented:** it's used when an email is received and the plugin has to comment a ticket. There is only one field to customize:
 - **Comment:** the comment that will be added to the issue

Special words available

When you use these words in your template they will be replaced with their corresponding value:

- **\$subject\$** – the subject of the received email.
- **\$body\$** – the body of the received email.

Outbound templates

Outbound Templates	
Issue Created Subject * <input type="text" value="Ticket no. \$ticketNo\$ created."/> Body * <div><div>Your ticket has been created successfully. You can add comments on this sending us an email with the ticket number in the subject.</div></div>	Issue Commented Subject * <input type="text" value="Ticket no. \$ticketNo\$ commented."/> Body * <div><div>Your ticket has been commented: \$comment\$</div></div>
Issue Resolved Subject * <input type="text" value="Ticket no. \$ticketNo\$ resolved."/> Body * <div><div>Your ticket has been resolved.</div></div>	Issue Commented Notification Subject * <input type="text" value="Ticket no. \$ticketNo\$ comment added"/> Body * <div><div>Your comment has been successfully added to your ticket no. \$ticketNo\$</div></div>
<div> <input type="button" value="Save templates"/> <input type="button" value="Default values"/> <input type="button" value="Cancel"/> </div>	

When an issue event is triggered and the plugin is about to send a notification email, the outbound templates are used, there are four outbound templates available to customize:

- **Issue Created:** it's used when a new ticket is being created in order to inform the user about the issue creation.
- **Issue Resolved:** it's used when a ticket is being resolved in order to inform the user about the issue resolution.
- **Issue Commented:** it's used when a ticket is being commented in order to inform the user about the comment that has just been added.
- **Issue Commented Notification:** it's used when a user comment is added to a ticket in order to inform the user that it was successfully added.

The available fields to customize are the same in all outbound templates:

- **Subject:** The subject of the notification email that is going to be sent.
- **Body:** The body of the notification email that is going to be sent.

Special words available

When you use these words in your template they will be replaced with their corresponding value:

- **\$ticketNo\$** – The ticket number associated with this issue.
- **\$summary\$** – The summary of the issue.
- **\$description\$** – The description of the issue.
- **\$comment\$** – The comment that has just been added to the issue (only available on "Issue Commented").

* Note: if you selected HTML emails in [General Settings](#) you can add html tags to the Outbound templates

Logging

In the Log Section of the configuration you will be able to find messages that might be useful to detect problems with the plugin.

Email Helpdesk Configuration

Services
Filters
Templates
General
Log
License

Log

Service Name	Log Level	Log Type	Date - Time	User	Email	Description	Show Email
--------------	-----------	----------	-------------	------	-------	-------------	------------

Log messages older than a week will be automatically deleted

Delete

License

In this section of the configuration you can review you license status and if your license is not valid you will find a link to the license administration page.

Email Helpdesk Configuration

Services
Filters
Templates
General
Log
License

License management

Your license is valid
You can manage your license [here](#)

License Details

Status	Valid developer evaluation license
Plugin Key	es.edataunited.supportSystemJiraPlugin.supportSystemJiraPlugin
Maximum Number of Users	10
Creation Date	September 10, 2112
Expiration Date	August 21, 2014
Maintenance Expiration date	August 21, 2014
Description	Test license for plugin developers
Organization	Atlassian
Server Id	BG9T-XUV4-KZZH-B01W
Support Entitlement Number (SEN)	

Working in JIRA with the plugin

Email Helpdesk issue panel

View Issue Panel

When you finish the plugin configuration, you will see on the issues created by the plugin a new panel (Email Helpdesk) on the right side of the view issue screen. The fields displayed in the Panel are detailed below:

- **Ticket #:** the ticket number associated with this issue.
- **Default Action:** the default action taken when a comment is added.
- **Expression to Send / not Send:** the expression that comments must start with in order to not trigger the default action. In the example screenshot, if the comment starts with "I:" then it will not be sent.
- **User email:** the email of the user that originated this issue.
- **Ticket Level:** the possible values for this field are 1, 2 or 3. If you wish to learn more about the different levels, please refer to the [Introduction Section](#).
- **Escalate Issue / De-Escalate Issue:** Buttons to move the tickets across the support levels

Email Helpdesk

Ticket #: MYPRODUCT-1
 Default Action: Send Comment
 Expression to not Send: I:
 User email: john.customer@example.com
 Ticket Level: 1
[Escalate Issue](#) [De-Escalate Issue](#)

Issue panel automatic actions

There are basically two automatic actions that the plugin will perform for you:

- Create issues/tickets when you receive emails in the address you have specified.
For example: If you configure the plugin to listen on address "support@example.com", and your first level support project is "TEST", then when you receive an email on "support@example.com" inbox an issue will be created on the "TEST" project. More information about how to configure the plugin is available at our [Jira Configuration](#) section and also our [Services](#) section.
- Send an alert email to the user who originated the issue/ticket when you resolve the issue.
For example: we are working on the issue TEST-1 which was created by our customer with email "customer@example.com", when we resolve issue "TEST-1" an email will be sent to "customer@example.com" informing about this status change. The notification sent can be configured at [Templates](#) configuration.

Communicating with the customer

Comments Area


The comments area is where the communication with the customer takes place, the comments sent to the customer are marked as "Public Comment". In order to make public or private comments we should take into account the information displayed in the Panel.

If a customer wants to add a comment, he has to send an email to the service mail address with the Ticket # in the subject of the mail and from the mail address that originally created the issue.

In the example, the Support Agent user added a comment (the default action is "Send Comment") and the comment was sent by a notification email to the customer. The customer replied the email (With the ticket number in the subject) and his comment was successfully added to the ticket.


Activity

[All](#)
[Comments](#)
[Work Log](#)
[History](#)
[Activity](#)

▼  **Support Agent** added a comment - 02/Jan/14 1:52 PM

----- Public Comment -----

Could you please give some more details?

▼  **External Users** added a comment - 02/Jan/14 1:54 PM

----- Public Comment -----

Of course,

When I try to log in, I receive an error: "invalid username"

—

John Customer

Escalating issues

In order to escalate/de-escalate an issue the buttons located in the [Email Helpdesk issue panel](#) should be used. When one of these buttons is clicked the target project where the issue should be moved to will be asked and also, optionally, a comment can be added to the issue at this screen.

Resolution: Unresolved

Escalate Issue

Project*

Comment

ssian JIRA Project Management Software (v6.3.4#6332-sha1:51bc225) · About JIRA · Report a prob